IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	§	Filed: April 14, 2004
Dettinger et al.	§	
	§	Group Art Unit: 2176
Serial No.: 10/824,064	§	
	§	Examiner: Nathan Hillery
Confirmation No.: 6073	§	

For: SIMPLIFIED AND OPTIMIZED PROCESS FOR APPLICATION USER INTERFACE TESTING AND VALIDATION

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Nathan Hillery, or electronically transmitted via EFS-Web, on the date shown below:

June 18, 2008 /Tammi Thomas/
Date Tammi Thomas

REPLY BRIEF

Dear Sir:

Applicants submit this Reply Brief to the Board of Patent Appeals and Interferences in response to Examiner's Answer dated April 18, 2008. While Applicants' maintain each of the arguments submitted in Applicants' previously submitted Appeal Brief, Applicants make the following further arguments in light of the Examiner's Answer. Please charge any additional fees that may be required to make this Reply Brief timely and acceptable to Deposit Account No. 09-0465/ROC920030346US1.

REMARKS

Claims 1-59 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Brook*, U.S. Pat. No. 2002/0038320

The Examiner continues to suggest that *Brook* discloses the method recited by claim 1 that includes:

> parsing, by a parser, two or more documents in tandem on an element-by-element basis, whereby the elements of each of the documents are sequentially parsed; [and]
> upon parsing each of the respective sequential elements in a first document of the two or more documents and each of the other documents, comparing the respective parsed elements to one another.

Claims 14, 31, 37, and 47 each recite a similar limitation. Respectfully, Applicants continue to disagree.

Brook is directed to a technique for reducing the memory requirements for parsing an XML document to determine whether it is both "well-formed" and "valid." For example, *Brook* provides:

> The inventive concept disclosed in this specification is based on the idea that memory requirements of an XML parser can be reduced, and various performance metrics can be improved, by performing a "perfect" hash of the XML tags, and possibly other elements within an XML file. ... This idea allows an arbitrary XML tag to be treated as a numeral or code, which can be stored in numeric form in memory. Since a parser normally preserves some portion of an XML structure in memory as the structure is parsed, conversion of XML tags to unique numerals allows memory requirements to be reduced, and furthermore, allows string-to-string comparisons to be replaced with equivalent, but much faster numerical comparisons.

That is, *Brook* discloses that replacing text-string markup tags with numeric values may reduce memory requirements, allowing numerical comparisons between elements during a parsing process. Consider the following example:

XML Fragment	Modified XML Fragment
<note> <to>Tove</to> <from>Jani</from> <heading>Reminder</heading> <body>Don't forget me this weekend</body>	<1> <2>Tove</2> <3>Jani</3> <4>Reminder</4> <5>Don't forget me this weekend<5> </1>

</note>	

In this example, the XML fragment includes the markup tags of <note>, <to>, <from>, <heading>, and <body>. *Brook* discloses parsing a document like this to identify the tags and to replace them with a numerical value (e.g., a hash of the tag's text-string). Thus, after processing according to the teachings of *Brook*, the above XML fragment could be replaced with the modified one.

Once this is done, the modified XML fragment may be parsed more efficiently using the numerical values than using the text-string based nodes of the original XML fragment. *Brook* describes that XML documents are often parsed to determine "well-formedness" and "validity. Additionally, *Brook* goes into significant detail describing two well-known concepts related to XML documents – "well-formedness" and "validity." And further, how the process of performing "Well-formedness" checks and "validation" checks by comparing markup tags formatted as text strings can consume significant resources. For example, *Brook* describes this problem as follows:

> Significant memory requirements arise from the verbose nature of the XML document, resulting in correspondingly significant memory requirements to store the document structure in its original string form. This document structure is referred to in the step 212. Furthermore, an associated significant processing load, relating to performance of string comparisons between variable length alpha-numeric strings, arises both in the well-formedness checking step 214, and in the validation checking step 220.

Brook, ¶ 219. As disclosed in Brook (and as is well known for XML generally):

> Well-formedness checks test the document for compliance with general structural rules, particularly whether tags in a document have been properly nested. ... "Validation checks involve a comparison of syntactic elements in a document against validity constraints defined in a Validation Reference Document (referred to as a VRD for the sake of brevity) such as a document type definition (DTD).

Brook, ¶ 0213.

The Examiner relies on a description of the "Well-formedness" checks and "validation" checks to argue that *Brook* discloses the claimed step of parsing, by a parser, two or more documents in tandem on an element-by-element basis, whereby the elements of each of the documents are sequentially parsed" and "upon parsing each

of the respective sequential elements in a first document of the two or more documents and each of the other documents, comparing the respective parsed elements to one another." For example, the Examiner suggests:

> (*Brook* teaches parsing two or more documents by elements and creating hash tables of the parsed elements for efficient comparisons. The hash tables are taught as a further step in parsing by elements for comparisons, and is taught as being preferable to a direct comparison by elements, however, the direct comparison is also taught as the inefficient method. See, *Brook*, paragraphs [0002] and [0206]-0225].
> *Brook* teaches parsing and comparison for purposes of comparison and validation, which is determining whether the documents are at least equivalent. See, Brook, paragraph [0236].

Examiner's Answer, p. 4. However, the Well-formedness" checks and "validation" checks relied on by the Examiner simply do not disclose the claimed step of "parsing ... two or more documents in tandem, ... comparing the respective parsed elements to one another, and on the basis of the comparison, determining whether the documents are at least equivalent."

A couple examples of "well-formed' and "valid" XML documents should clearly illustrate the distinction. First, as discussed above, *Brook* teaches that text string mark-up tags may be replaced with numerical values (e.g., a hash of the text-string) to increase the efficiency of parsing operations. Once replaced, a document may be parsed to determine whether it is "well formed." The following table illustrates an example of improper nesting using XML, i.e., of a document that is not "well-formed:"

Well formed	Not well formed XML Fragment
<1> <2>Tove <3> Jani </3> </2> <4>Reminder</4> <5>Don't forget me this weekend<5> </1>	<1> <2>Tove <3> Jani </2> </3> <4>Reminder</4> <5>Don't forget me this weekend<5> </1>

In the example above, the XML on the left is not well-formed since the <3> element is opened inside the <2> element, but is not closed inside the <2> element. Other rules for "well-formedness" for XML documents include the following:

> XML Elements Must Have a Closing Tag
> XML Tags are Case Sensitive
> XML Documents Must Have a Root Element

XML Attribute Values Must be Quoted

The parsing process for "well-formedness" may be used to verify that a particular XML document conforms to these rules. At the same time, nothing in this process, even when performed using numerical values for node names as taught in *Brook*, discloses the claimed step of "parsing ... two or more documents in tandem, ... comparing the respective parsed elements to one another, and on the basis of the comparison, determining whether the documents are at least equivalent."

Similarly, the process of determining whether an XML document is "valid" does not disclose this limitation. A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD) (or "VRD" as used in *Brook*). Another example should clarify this distinction. Assume the "notes" documents above are composed using the following example "notes" DTD:

Document DTD for a "note"

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

Like the nodes of a document itself, *Brook* teaches that elements of the DTD may be replaced with numerical values. Thus, consistent with the examples above, the DTD for the "note" could be modified with numerals as follows:

Modified Document DTD for a "note"

```
<!ELEMENT 1 (2, 3, 4, 5)>
<!ELEMENT 2 (#PCDATA)>
<!ELEMENT 3 (#PCDATA)>
<!ELEMENT 4 (#PCDATA)>
<!ELEMENT 5 (#PCDATA)>
```

Once modified, a given "note" XML document ma be evaluated to determine whether it conforms to the "note" DTD. The first line specifies that a note may contain a "1" node (corresponding to "note" node of the unmodified DTD), and that the "1" node may contain other nodes of type "2", "3", "4", and "5." The remaining lines specify that the "2", "3", "4", and "5" nodes (corresponding to the <to>, <from>, <heading>, and <body> nodes) may contain parsed character data. This DTD may be used to determine whether different documents of type "note" (or "1") are valid.

For example, consider the following "note" XML documents:

Valid XML Fragment	Invalid XML Fragment	Invalid XML Fragment
<1> <2>Tove</2> <3>Jani</3> <4>Reminder</4> <5>Don't forget me this weekend<5> </1>	<1> <2>Tove <3>Jani</3></2> <4>Reminder</4> <5>Don't forget me this weekend<5> </1>	<1> <2>Tove</2> <4>Reminder</4> <5>Don't forget me this weekend<5> <6>Date:7/4/2008</6> </1>

The first "note" conforms to the note DTD, where the other two "notes" do not. The note in the middle column is invalid because it includes a <3> element nested inside the <2> element, where the DTD specifies that the <2> element may contain parsed character data. The note in the right column is invalid because it includes a <6> element where the DTD specifies that the <1> element may contain <2>, <3>, <4>, and <5>, elements. *Brook* teaches that the efficiency of the process for validating and XML documents such as these may be improved by converting the text-based node names to numerical values for processing. Doing so replaces string-to-string comparisons with numerical comparisons, which are typically performed much more efficiently. Applicants submit that this process is plainly distinct from the limitations recited by claim 1.

Although the Examiner cites *Brook,* ¶ 236 as teaching the claimed step of "upon parsing each of the respective sequential elements in a first document of the two or more documents and each of the other documents, comparing the respective parsed elements to one another," the passage provides an example of XML document validity:

> If, on the other hand, no error is detected, the parsing process 344 is directed to the optional process 348, in which the validation checking step 326, using respective processors 414 or 505, is performed with reference to a DTD or an XML Schema. As noted, validation checking is a more detailed form of checking than well-formedness checking. Thus, for example, whereas the well-formedness check considers whether the "Hamlet" tag pair is properly nested within the "Shakespeare" tag pair, validity checking, in contrast, both checks for proper nesting in the sense that the "Hamlet" tag pair is fully nested within the "Shakespeare" tag pair, but also checks whether "Hamlet" tag pairs may legally be nested in this way. There may, for example, be a situation where, in fact, "Shakespeare" tag pairs must be nested within "Hamlet" tag pairs, rather than the other way around. Thus, the validity checking process checks hierarchical relationships of tags, in this case being whether "Hamlet" tag pairs may be

nested within "Shakespeare" tag pairs, as well as considering whether nesting has been properly, namely completely, performed.

Brook, ¶ 236

Nothing in this discussion of a "well-formed" XML document using tags with Shakespearian names discloses the claimed steps of "parsing … two or more documents in tandem, … comparing the respective parsed elements to one another, and on the basis of the comparison, determining whether the documents are at least equivalent." Instead, this passage discloses an example of when a document may be "well-formed."

Lastly, consider the following example of valid note documents (represented with numerals for node names as disclosed in *Brook*)

Valid XML fragment	Valid XML Fragment
<1> <2>Tove </2> <4>Reminder</4> </1>	<1> <3>Jani</3> <5>Don't forget me this weekend<5> </1>

Both of these XML fragments would be determined to be valid according to the methods disclosed in *Brook*. In particular, the each document is individually compared against the "note" DTD to determine if each "note" is individually valid. In this example, the node names have been replaced with numerals to increase the processing efficiency of these comparisons, as taught by *Brook*. Further, the process (and any comparisons performed thereby) of Brook would conclude that the two XML fragments are both well formed and valid. At the same time, they are clearly not equivalent as they contain substantially different elements. Thus, Applicants submit that the passages from Brook relied on by the Examiner do not, in fact, disclose the claimed limitation of "parsing … two or more documents in tandem, … comparing the respective parsed elements to one another, and on the basis of the comparison, determining whether the documents are at least equivalent. For example, at no point would the two fragment listed above be compared against one another. Instead, as should be clear from the discussion above, each document is parsed and validated against the "note" DTD, albeit more efficiently using numerical comparisons instead of string-based comparisons.

Accordingly, for all the foregoing reasons, Applicants submit that Brook does not disclose the limitations of claim 1, as suggested by the Examiner. Similarly, Applicants submit that *Brook* does not disclose the limitations of independent claims 14, 31, 37, and 47, along with the dependent claims. Accordingly, Applicants respectfully request that the Board vacate the rejection of these claims and the claims dependent therefrom, and direct that these claims be allowed.

Further, regarding claim 14, Applicants submit that *Brook* does not disclose a "method of testing and validating user interface content," recited by this claim. In particular, *Brook* does not disclose the claimed steps of:

> submitting a request to an application;
> in response to the request, receiving a response document from the application retrieving from storage a control document previously returned from the application in response to the request;
> sequentially determining each element of the response document and the control document; and
> for at least some of the respective sequentially determined elements from the respective documents, comparing the elements to one another.

The Examiner argues the following with respect to claim 14:

> *Brook* teaches parsing two or more documents by elements and creating hash tables of the parsed elements for efficient comparisons. The hash tables are taught as a further step in parsing by elements for comparisons, and is taught as being preferable to a direct comparison by elements, however, the direct comparison is also taught as the inefficient method. See, *Brook*, paragraphs [0002] and [0206]-0225]. *Brook* teaches parsing and comparison for purposes of comparison and validation, which is determining whether the documents are at least equivalent. See, *Brook*, paragraph [0236]. *Brook* also teaches parsing two documents element by element and comparing the documents for validation. See, *Brook*, paragraphs [0060]-[0069]. See, *Brook*, paragraphs [0014]-[0274], teaching comparison of a parsed document against a control document, which is taught as a Validation Reference Document (VRD).

Examiners Answer, p.12. Despite the distinct limitations of Claim 14, the non-underlined portion of this rejection simply repeats the rejection of Claim 1. Further, the underlined portion suggests that the Validation Reference Document (VRD) reference document discloses the claimed "control document" which is compared with the response document to identify whether elements are equivalent to one another.

Applicants strongly disagree. As demonstrated above, the "Version Reference Document" disclosed in *Brook* provides an XML DTD modified to have text references to tag names replaced with numeric values. By its own terms, *Brook* describes the VRD as follows:

> Validation checks involve a comparison of syntactic elements in a document against validity constraints defined in a Validation Reference Document (referred to as a VRD for the sake of brevity) such as a document type definition (DTD), as described in Section 5.1 of the aforementioned W3C Recommendation. DTDs and XML Schemas are examples of VRDs against which validation checks can be performed.

Brook , ¶ 213. However, Claim 14 recites

> sequentially determining each element of the response document and the control document;
> for at least some of the respective sequentially determined elements from the respective documents, comparing the elements to one another; and
> on the basis of the comparison, determining whether the elements are equivalent.

That is, the claim specifies to compare at least some elements of the control document and the response document with one another and, "on the "determining whether the elements are equivalent." In sharp contrast, the VRD provides a set of "validity constraints" evaluated to determine whether a given document complies with the constraints. Plainly, it would make no sense to compare elements of the VRD with a given document to determine if they are equivalent. Consider the following Example VRD for a "note" document:

VRD for a "note"	XML Fragment
<!ELEMENT 1 (2, 3, 4, 5)> <!ELEMENT 2 (#PCDATA)> <!ELEMENT 3 (#PCDATA)> <!ELEMENT 4 (#PCDATA)> <!ELEMENT 5 (#PCDATA)>	<1> <2>Tove</2> <3>Jani</3> <4>Reminder</4> <5>Don't forget me this weekend<5> </1>

Clearly the "VRD" for the "note" document is not at all equivalent to the XML fragment. At the same time, the XML fragment is clearly valid, based on the constraints specified by the VRD for the "note" document.

Accordingly, for all the foregoing reasons, Applicants submit that claim 14, as

well as the respective dependant claims, are allowable, and Applicants respectfully that the Board reverse the final rejection.

CONCLUSION

The Examiner errs in finding that claims 1-59 are unpatentable over *Brook* under 35 U.S.C. § 103(a).

Withdrawal of the rejection and allowance of all claims is respectfully requested.

Respectfully submitted, and

S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/
Gero G. McClellan
Registration No. 44,227
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd., Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellants